# Reduction of Target Fault List for Crosstalk-Induced Delay Faults by using Layout Constraints *

Keith J. Keller, Hiroshi Takahashi†, Kim T. Le‡, Kewal K. Saluja, Yuzo Takamatsu†

*University of Wisconsin - Madison, U.S.A*
{*kellerk, saluja*}*@ece.wisc.edu*

† *Ehime University, Japan*
{*takahashi, takamatsu*}*@cs.ehime-u.ac.jp*

‡ *University of Canberra, Australia*
*kiml@ise.canberra.edu.au*

## Abstract

*We propose a method of identifying a set of crosstalk induced delay faults which may need to be tested in synchronous sequential circuits. During the fault list generation 1) we take into account all clocking effects, and 2) infer layout information from the logic level description. With regard to layout constraints we introduce two methods, namely the distance based layout constraint and the cone based layout constraint. The lists of the target faults obtained by the proposed methods are substantially smaller than the sets of all possible combinations of faults.*

## 1 Introduction

Circuit testing is an important part of the circuit design and manufacturing process. Traditionally, simple classical faults models, such as stuck-at-0 or stuck-at-1, were used to test for faults. However, with the scaling of VLSI features in synchronous sequential circuits, testing for non-classical faults, such as crosstalk faults, has become an important problem [2, 3, 4, 5, 6, 7, 9]. In this paper we address the crosstalk faults issue in synchronous sequential circuits.

A crosstalk-induced delay fault is produced between a pair of lines, an aggressor line (A-line) and a victim line (V-line). This delay is induced due to simultaneous, or near simultaneous, transitions on both lines. These unexpected changes in the signal propagation time can cause faulty behavior. Therefore, it is necessary to generate tests for crosstalk faults between A-lines and V-lines. However, the number of all possible combinations of A-lines and V-lines is very large and impractical to deal with. Therefore, we have proposed a method of identifying a set of crosstalk-induced delay faults which may need to be tested in synchronous sequential circuits without using any layout information [8]. Reduction in the size of the target faults is also useful for diagnosing the crosstalk faults [10]. The strength of the crosstalk effect between an A-line and a V-line depends on their physical closeness, therefore circuit layout must be considered in reducing the crosstalk fault list. When layout information is not available a good estimation of gate locality may be made using our "distance" and "cone" constraint methods. Finally, if the layout information is indeed available, such information can be used to further reduce the fault list generated by our method.

Further, in our previous work [8] we did not address all cases dealing with clock transitions and layout constraints. Therefore, the fundamental contributions of this paper are as follows:

1) We consider the effect of all clock edges of an aggressor clock line to identify target crosstalk faults.

2) We use logic connectivity to deduce layout information and reduce the number of candidate faults.

This paper is organized as follows. For completeness, in section 2 we give a brief overview of our previous work. In section 3 we consider the impact of the ineffective edge of the aggressor clock line to identify target crosstalk faults, and propose two methods by using layout connectivity to reduce the number of candidate faults. In section 4 we present experimental results on ISCAS'89 and ITC'99 benchmark

*This research was partially supported by the National Science Foundation grant MIP9714034. This work was performed when Prof. Takahashi and Prof. Le were visiting UW-Madison.

Proceedings of the 11 th Asian Test Symposium(ATS'02)
1081-7735/02 $17.00 © 2002 IEEE

IEEE COMPUTER SOCIETY

circuits to evaluate our methods. Finally, we conclude the paper.

## 2 Our Previous Work

For completeness of this paper we review our previously reported work. However, there are a number of updates we have made to our earlier work and these are also identified here. This section starts by reviewing our previous assumptions and dividing crosstalk faults into four different cases, and finishes with a classification of crosstalk faults.

We use the slow-fast-slow clock method [1] to test crosstalk-induced delay faults in sequential circuits. The following assumptions were made while dividing the crosstalk faults into four different cases [8]:
**Assumption 1:** We do not consider clock-skew; we assume positive edge triggered flip-flops.
**Assumption 2:** The clock cycle for the circuit is determined by the delay of the structurally longest path in the circuit.
**Assumption 3:** A crosstalk-induced speedup or slowdown on a V-line is excited if the transition on an A-line occurs within the $\Delta$ timing window of a transition on the V-line [3, 4, 5, 6].
**Assumption 4:** The size of extra delay caused by the crosstalk-induced delay fault is one gate delay.
**Assumption 5:** The setup time of the flip-flop is one unit delay.

In [8] the term "ineffective clock edge" was used while computing the number of crosstalk faults. We give the formal definition of this term for clarification of the term and why this condition needs to be relaxed.
**Definition** (**ineffective clock edge**): An edge of the clock on which the flip-flops do not latch data is called the *ineffective clock edge*.

Note that for a positive (negative) edge trigger design the falling (rising) edge is the ineffective clock edge. In this paper we correctly account for the faults that may be caused by the ineffective clock edge.

The candidate A-line and V-line for crosstalk faults are clock-lines, primary inputs, and gate outputs. For simplicity we refer to the primary inputs and the gate outputs as "lines", and clock signals as "clock-lines". The interactions between A-line / V-line pairs are divided into the following four cases. An in-depth review of each of these cases can be found in [8].

Case 1) A-line and V-line are lines.
Case 2) A-line is a line and V-line is a clock-line.
Case 3) A-line is a clock-line and V-line is a line.
Case 4) A-line and V-line are clock-lines.

## 3 Timing and Topological Information For Identifying Target Faults

In our previous paper [8] we did not consider the effect of an ineffective clock-edge of an aggressor clock line. In this paper we refine our previous conditions for target crosstalk faults (i.e. if the positive clock edge triggers the Flip-Flops, the falling edge of the clock will be considered the ineffective edge of the clock) to address this issue. Below is the classification of each case with Case 3 being revised since [8].

### 3.1 Classification

**Case 1 [8])** A-line and V-line are lines - if the following conditions are satisfied the A-line and V-line pair is a target crosstalk fault.
C1) The V-line must be included in a longest path.
C2) The relationship between $t\_ag$ and $t\_vi$ satisfies the following inequality:

($t\_vi$ - $\Delta$ unit delays) $\leq t\_ag \leq$ ($t\_vi$ + $\Delta$ unit delays)

where $t\_ag$ and $t\_vi$ represent the transition times at A-line and V-line, respectively.

**Case 2 [8])** A-line is a line and V-line is a clock-line - this pair can not satisfy the condition for exciting a crosstalk induced delay fault.

**Case 3 - Updated)** A-line is a clock-line and V-line is a line - if the following conditions are satisfied the V-line and A-line pair is a target crosstalk fault.
C3) The V-line must be included in a longest path.
C4) The relationship between $t\_ag$ and $t\_vi$ satisfies the following inequality.

($t\_vi$ - $\Delta$ unit delays) $\leq t\_ag \leq$ ($t\_vi$ + $\Delta$ unit delays)

**Case 4 [8])** A-line and V-line are clock-lines - if the following condition is satisfied the A-line and V-line pair is a target crosstalk fault.
C5) The fan-in of the flip-flop with the victim clock-line is included in a longest path.

The complete classification of all faults is shown in Figure 1.

In our methods we consider both the topological and the timing information. Regarding timing information we calculate the latest transition time and the earliest transition time at each line and find the lines included in the longest paths. We use the timing information in order to determine whether an A-line and V-line pair satisfies the conditions for the target crosstalk fault. Under our timing model we


IEEE COMPUTER SOCIETY

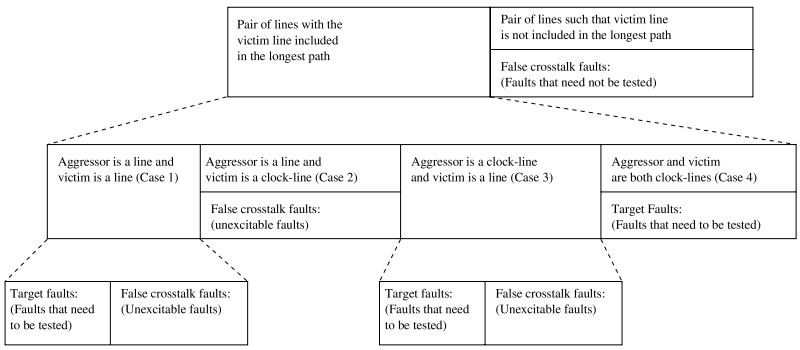


**Figure 1. Relationship between target crosstalk and false crosstalk fault.**

assume that each gate has a delay of one time unit (unit delay model).

**Definition (latest transition time):** The *latest transition time (ltime)* at a line is the maximum delay on any path from any primary input or the output of a flip-flop to this line.

**Definition (earliest transition time):** The *earliest transition time (etime)* at a line is the minimum delay on any path from any primary input or the output of a flip-flop to this line.

In our study, an aggressor line has a timing window of [*etime* of aggressor line, *ltime* of aggressor line]. Similarly, the victim line has a timing window of [(*ltime* of victim line - $\Delta$ unit delays), (*ltime* of victim line + $\Delta$ unit delays)], and $\Delta$ is assumed to be 1 or 2 unit delays. This assumption is justified by the experimental results reported in [3, 4, 5, 6]. However, for simplicity we describe the procedure given in this section for $\Delta$ to be one unit delay. In order to identify the target faults we check whether the timing window at A-line overlaps with the timing window at V-line.

## 3.2 Identification of Target Crosstalk Faults

The proposed method consists of two phases. In Phase 1, under the unit delay model, the algorithm calculates an *etime* and a *ltime* at each line. Next, the algorithm generates the $Set\_V$ that consists of all gate outputs, primary inputs, and flip-flops in longest paths by using the timing and topological information. In Phase 2, the algorithm determines whether the timing window for a V-line overlaps with the timing window of an A-line using the timing information. The algorithm and its two phases are described below.

**[Algorithm for identifying target crosstalk faults]**

**Phase 1:**

Step 1) Calculate the *latest transition time (ltime)* and the *earliest transition time (etime)* at each line.

Step 2) Using the maximum value of the *ltime*, identify the longest paths in the circuit.

Step 3) Identify the lines that are included in the longest paths by the topological information. Add these lines to the set of V-lines ($Set\_V$).

**Phase 2:**

Step 1) Select a V-line from the $Set\_V$ and remove the selected V-line from the $Set\_V$.

Step 2) Compare the timing window at the selected V-line and the *etime*s and the *ltime*s of other gate outputs considered as A-lines. For each of these A-lines, if the timing window [*etime*, *ltime*] of A-line overlaps with the timing window [(*ltime* - 1), (*ltime* + 1)] of V-line add the selected A-line and V-line pair to the list of the target crosstalk faults.

Step 3) If the $Set\_V$ is not empty go to step 1.

Step 4) Identify the clock-lines of flip-flops whose fan-in lines are included in the longest paths; these fan-in lines belong to the set of V-lines ($Set\_V$).

Step 5) Add the pairs (V-lines and other clock-lines) to the list of the target crosstalk faults.

## 3.3 Methods to Reduce Target Fault Lists Using Layout Constraints

Layout of the circuit plays an extremely important role in crosstalk fault excitation. Therefore, we propose two methods for determining A-line/V-line locality without a priori layout knowledge of the circuit.

### 3.3.1 Distance Layout Constraint

The distance of an A-line from a V-line is determined by the number of gate levels between the two lines. Suppose that a line $\alpha$ under consideration has a distance of D from the victim line. All inputs and fan-ins of the gate driving line


1081-7735/02 $17.00 © 2002 IEEE

IEEE COMPUTER SOCIETY

$\alpha$ have a distance of D+1; all fan-outs of the gates driven by line $\alpha$ also have a distance of D+1. Before describing an algorithm for calculating distance, we define several terms: the *victim set (VS)* is the set of all the victim lines within a given circuit. The *aggressor set (AS)* is the set of all possible aggressor lines to a particular victim line. The *driving gate* is the gate whose output is connected to the line of interest($\alpha$); the *driven gates* are the gates which have $\alpha$ as an input. Further, we assume that a line may have only one driving gate, but more than one driven gates.

The algorithm for the distance layout constraint method is described below.

**[Establishing a layout constraint of Distance 1 to n]**
For every victim line in VS do:

Step 1) Assign the victim line a distance of D = 0, and consider it as the line of interest.

Step 2) For each line of interest with distance D, identify its driving gate and its driven gates. All lines connected to the inputs of the driving gate and to the output of the driven gate which have not been assigned distance values are lines at distance D+1. Add these lines to the AS and consider them as lines of interest.

Step 3) Repeat Step 2) until D = Threshold.

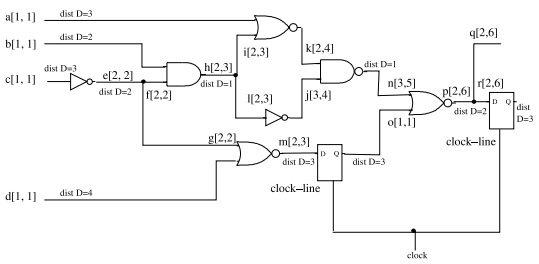


**Figure 2. Distance constraint method.**

We demonstrate the operation of the algorithm on the sequential circuit shown in Figure 2.
1. Let, line j be the initial line of interest (i.e. D(j) = 0).
2. Identify the driving gate (inverter j) and driven gate (NAND n); lines with D= 1 are h(i, l) and n. All these lines are added to the set AS.
3. Lines with D = 2 are b, e(f, g), k(twice), and p(q, r).
4. In our example the largest distance found is 4 on line d.

In Figure 2 line j is the V-line and the distances of all possible A-lines from line j are shown. In this example all possible A-lines are included in a constraint of distance 4.

Under the distance layout constraint of distance 2, lines b, e, h, k, n and p are added to AS of the victim line j. We only check whether the A-line/V-line pairs (b-j, e-j, h-j, k-j, n-j and p-j) satisfy the timing condition or not.

### 3.3.2 Cone Layout Constraint

The Cone layout constraint method is another attempt to determine A-line locality to a potential V-line without a priori knowledge of the circuit layout.

The first phase of a V-line cone is constructed by finding the cone of output gate lines from the V-line to a "$ltime$" boundary defined by the V-lines [$etime$, $ltime$] timing window; the boundary includes all possible aggressor lines with an earliest transition time ($etime$) equal to the latest transition time ($ltime$) of the V-line. The second phase of the cone construction begins with all the gate lines lying on the $ltime$ boundary; all the inputs to these gates are included in the cone set ending at the $etime$ boundary. Every subsequent cone distance begins from the $etime$ boundary of gates included in the previous cone set. An algorithm of the cone layout constraint method is described below.

**[Establishing a layout constraint cone]**

Step 1) Choose a victim line from VS as the initial line of interest with D = 0 (where D is called the conic distance), and determine its [etime, ltime] transition window.

Step 2) Find the fan-out cone of the line of interest; terminate searching (forward) when a line with an etime $\geq$ the ltime of the victim line is reached; the line at this boundary is called a *leaf-line* of the fan-out cone.

Step 3) For each *leaf-line* of the fan-out cone, find its fan-in cone; terminate the backward search when a line with a ltime $\leq$ the etime of the victim line is approached; a cone with conic distance D=1 is formed from all lines that belong to a fan-in cone of a *leaf-line* line. This cone is called the 1st cone.

**[All subsequent Cones from Distance 2 to n]**

Step 4) For creating the (D+1)th cone consider each line in the (D)th cone as a line of interest. Now Steps 2) and 3) are repeated for each of these lines and the obtained cone is the (D+1)th cone.

Step 5) The AS is all the lines included in the highest order cone.

We illustrate the method for establishing cones of distance 1 to n for an abstract representation of a circuit shown in Figure 3. We note:
1. the figure identifies a victim line shown as the output of a gate
2. the stopping boundary after Step 2) is labeled "$ltime$ boundary of [$ltime$, x]"
3. the Cone of Distance 1 with respect to the victim gate is shown in light shading.


IEEE COMPUTER SOCIETY

The second phase is to construct the fan-in cone for every line in the ltime boundary found in the first phase. The construction is a back-trace process. For each path the back-trace stops when it arrives at a line with ltime $<$ the victim etime (etime boundary).

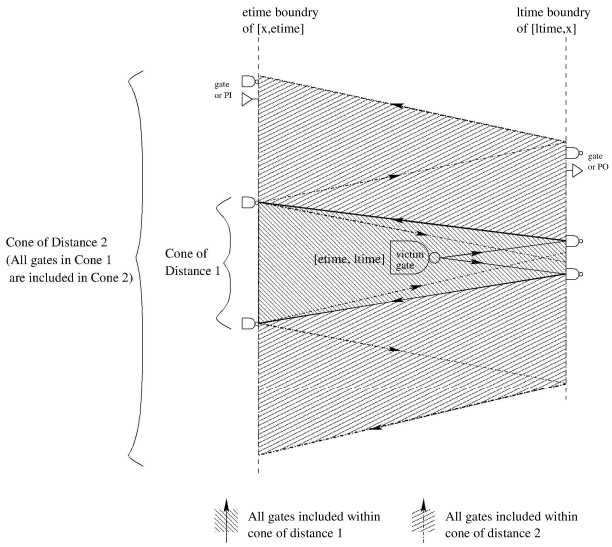


**Figure 3. Cone constraint method.**

## 4 Experimental Results

We implemented the algorithms described in section 3 for identifying target crosstalk-induced delay faults in sequential circuits in C code. We conducted experiments on ISCAS'89 and ITC'99 benchmark circuits as follows. First we identify target faults for ISCAS'89 and ITC'99 circuits (Table 1). In our experiment we use a $\Delta = 1$. We also looked at "the ineffective clock edge" to estimate the growth of the number of target faults that belong to Case 3. Additionally, we vary the duty cycle of the clock. Results for the two constraint methods are given in Table 2.

When looking at either of these reduction methods one must realize that this is an attempt to help determine physical layout information, from the logic design, when no physical layout information is available. However, if physical layout information is available it should be used in conjunction with our reduction methods to produce more accurate results. One must also note that when using the "distance" constraint method described in this paper, the minimum distance of 2 is required. The reason being that the algorithm given in this paper defines all lines driving the gate whose input is the victim line to be at a distance of 2 (see Figure 2) from the victim line. It is for this reason the results presented in Table 2 use a constraint distance of 2.

In Table 1 the column "all possible pairs" contains all combination pairs of the gate outputs, primary inputs and clock-lines of the flip-flops. The column "number of candidate pairs" contains the total number of combinations between lines that are included in the longest paths and all other lines in the circuit. The columns "No. of target faults (duty cycle 50%)" and "No. of target faults (duty cycle 75%)" contain the total number of target faults for these cases respectively, that are obtained by our method taking into consideration the ineffective edge o f the aggressor clock-line. Information from Table 1 shows that the longest path information is useful to reduce the number of target crosstalk faults.

For example, let us consider the circuit s38584 in Table 1. The total number of A-line and V-line pairs is 429,173,372. The total number of candidate pairs is reduced to 4,056,744 in phase 1. In Phase 2 the set reduces further with 316,185 faults identified to belong to the list of the target crosstalk faults in Case 1, 7 faults in Case 3 (duty cycle 50%), and 1,451 faults in Case 4. Thus 3,454,127 faults in Case 1 are false crosstalk faults, 20,717 faults in Case 2, and 264,257 faults in Case 3 are false crosstalk faults. Hence, we identify 317,643 target crosstalk faults for s38584 circuit. The percentage of target crosstalk faults for s38584 circuit is $8\%$ of the total number of candidate pairs.

We note that the number of target faults for s15850, s35932 and s38584 circuits, which have very large number of candidate pairs, reduces substantially by using our method as well. In this experiment we also vary the size of the duty cycle of the clock. We compare the results in Table 1 with those reported in [8] to determine the effect of the ineffective edge of aggressor clock lines. Experimental results show that the number of target faults identified in Case 3 is small. Also, there is not a significant growth in the number of target faults when the duty cycle of the aggressor clock line is varied. Results for ITC'99 benchmark circuits are very similar to those for the ISCAS'89 circuits.

Table 2 shows the effectiveness of the Distance and Cone constraint methods. The column "No. of target faults" contains the target crosstalk faults for Cases 1, 3, and 4. The columns "No. of target faults (Dist 2)" and "No. of target faults (Cone 2 )" contain the same target crosstalk faults for the three target cases as before, but using the respective layout constraint methods described in section 3.3. The percentage of target faults is the ratio of the number of target faults over the number of candidate faults. Taking another look at s38584 one will notice that the percentage of target crosstalk faults is reduced form $8\%$, without layout constraint s, to $2\%$ using a Cone constraint of distance 2 and reduced to a mere $0.13\%$ using a Distance layout constraint of distance 2. Results for ITC'99 benchmark circuits are again very similar to the results for the ISCAS'89 circuits.


1081-7735/02 $17.00 © 2002 IEEE

IEEE COMPUTER SOCIETY

**Table 1. Identified crosstalk faults for varying sizes of duty cycle.**

| Circuit names | No. of target faults duty cycle (50%) | No. of target faults duty cycle (75%) | All possible pairs | No. of candidate pairs |
|---|---|---|---|---|
| s5378 | 53,253 | 53,254 | 8,955,056 | 221,970 |
| s9234 | 347,860 | 347,888 | 34,146,492 | 2,258,412 |
| s13207 | 107,000 | 106,994 | 74,831,150 | 1,546,954 |
| s15850 | 102,538 | 102,516 | 107,796,306 | 3,765,797 |
| s35932 | 22,105,636 | 22,105,892 | 317,819,756 | 196,019,320 |
| s38417 | 124,901 | 124,908 | 568,464,806 | 1,910,850 |
| s38584 | 317,643 | 317,639 | 429,173,372 | 4,056,744 |
| b01 | 247 | 248 | 1,892 | 629 |
| b02 | 250 | 250 | 702 | 510 |
| b03 | 2,240 | 2,261 | 20,880 | 9,396 |
| b04s | 3,857 | 3,862 | 346,332 | 27,534 |
| b06 | 335 | 334 | 2,862 | 682 |
| b07s | 2,421 | 2,421 | 169,332 | 17,940 |
| b08 | 1,067 | 1,067 | 26,406 | 4,209 |
| b09 | 5,687 | 5,699 | 24,806 | 13,320 |
| b10 | 1,483 | 1,490 | 33,306 | 5,174 |
| b11s | 8,411 | 8,411 | 225,150 | 30,300 |
| b12 | 11,166 | 11,167 | 1,059,870 | 62,221 |
| b13s | 2,004 | 2,006 | 107,912 | 7,620 |
| b14s | 33,528 | 33,532 | 22,283,120 | 918,525 |
| b15s | 73,680 | 73,680 | 77,836,506 | 2,030,000 |
| b17s | 534,615 | 534,615 | 580,641,312 | 18,210,000 |
| b20s | 63,811 | 63,811 | 88,284,212 | 1,770,000 |
| b21s | 66,522 | 66,522 | 95,658,180 | 1,838,000 |
| b22s | 102,975 | 102,979 | 226,457,352 | 2,857,000 |

**Table 2. Comparison of Distance and Cone constraint methods for ISCAS'89 and ITC'99.**

| Circuit names | No. of target faults | No. of target faults (Dist 2) | % target faults | No. of target faults (Cone 2) | % target faults |
|---|---|---|---|---|---|
| s5378 | 53,253 | 284 | 0.13 | 15,378 | 6.9 |
| s9234 | 347,860 | 1,516 | 0.07 | 106,282 | 4.7 |
| s13207 | 107,000 | 1,447 | 0.09 | 23,224 | 1.5 |
| s15850 | 102,538 | 1,751 | 0.05 | 29,299 | 0.78 |
| s35932 | 22.11M | 644,070 | 0.33 | 1.094M | 0.56 |
| s38417 | 124,901 | 514 | 0.03 | 21,995 | 1.2 |
| s38584 | 317,643 | 5,184 | 0.13 | 81,777 | 2.0 |
| b01 | 247 | 96 | 15.3 | 241 | 38.3 |
| b02 | 250 | 130 | 25.5 | 220 | 43.1 |
| b03 | 2,240 | 577 | 6.1 | 2,204 | 23.5 |
| b04s | 3,857 | 374 | 1.36 | 3,391 | 12.3 |
| b06 | 335 | 128 | 18.8 | 299 | 43.8 |
| b07s | 2,421 | 276 | 1.54 | 2,169 | 12.1 |
| b08 | 1,067 | 180 | 4.28 | 993 | 23.6 |
| b09 | 5,687 | 575 | 4.32 | 5,395 | 40.5 |
| b10 | 1,483 | 303 | 13.9 | 1,293 | 59.5 |
| b11s | 8,411 | 832 | 2.75 | 8,299 | 27.4 |
| b12 | 11,166 | 1,014 | 1.63 | 7,245 | 11.6 |
| b13s | 2,004 | 282 | 3.70 | 1,111 | 14.6 |
| b14s | 33,528 | 1,305 | 0.14 | 32,691 | 3.6 |
| b15s | 73,680 | 2,403 | 0.12 | 68,637 | 3.4 |
| b17s | 534,615 | 9,618 | 0.053 | 186,287 | 1.0 |
| b20s | 63,811 | 840 | 0.05 | 56,149 | 3.2 |
| b21s | 66,522 | 840 | 0.046 | 58,530 | 3.2 |
| Number of target faults includes cases 1, 3 and 4 | | | | | |
| M = $10^6$ | | | | | |

## 5 Conclusion

In this paper we proposed a method to identify target crosstalk faults using topological and timing information. We then proposed methods to reduce the number of target faults using two layout constraints. Our proposed methods can be very effective in reducing the target crosstalk fault list. In the experimental results for ISCAS'89 and ITC'99 benchmark circuits we show that the lists of the target crosstalk faults obtained by the proposed methods are significantly smaller than the set of all possible combinations of faults. Our proposed methods with layout constraints give excellent results in reducing the target fault lists.

## References


[1] P. Agrawal, V. D. Agrawal, and S. C. Seth, "Generating Tests for Delay Faults in Nonscan Circuits," in *IEEE Design & Test of Computers*, 1993, pp. 20–28.

[2] P. Chen, D. A. Kirkpatrick, and K. Keuzer, "Switching Window Computation for Static Timing Analysis in Presence of Crosstalk Noise," in *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 2000, pp. 331–337.

[3] W. Y. Chen, S. K. Gupta, and M. A. Breuer, "Analytic Method for Crosstalk Delay and Pulse Analysis Under non-Ideal Inputs," in *Proc. Int. Test Conf.*, Oct. 1997, pp. 809–818.

[4] W. Y. Chen, S. K. Gupta, and M. A. Breuer, "Test Generation in VLSI Circuits for Crosstalk Noise," in *Proc. Int. Test Conf.*, Oct. 1998, pp. 641–650.

[5] W. Y. Chen, S. K. Gupta, and M. A. Breuer, "Test Generation for Crosstalk-Induced Delay in Integrated Circuits," in *Proc. Int. Test Conf.*, Oct. 1999, pp. 191–200.

[6] W. Y. Chen, S. K. Gupta, and M. A. Breuer, "Test Generation for Crosstalk-Induced Delay Faults: Framework and Computational Results," in *Proc. Asian Test Symp.*, 2000, pp. 305–310.

[7] N. Itazaki, Y. Idomoto, and K. Kinoshita, "A Fault Simulator Method for Crosstalk Faults in Synchronous Sequential Circuits," in *Proc. Fault-Tolerant Comp. Symp.*, 1996, pp. 568–577.

[8] K. J. Keller, H. Takahashi, K. K. Saluja, and Y. Takamatsu, "On Reducing the Target Fault List of Crosstalk-Induced Delay Faults in Synchronous Sequential Circuits," in *Proc. Int. Test Conf.*, 2001, pp. 38–43.

[9] D. A. Kirkpatrick and A. L. Sangiovanni-Vincentelli, "Techniques for Crosstalk Avoidance in the Physical Design of High-Performance Digital Systems," in *Proc. Design Automation Conf.*, 1994, pp. 616–619.

[10] H. Takahashi, M. Phadoongsidhi, Y. Higami, K. K. Saluja, and Y. Takamatsu, "Simulation-based Diagnosis for Crosstalk Faults in Sequential Circuits," in *Proc. Asian Test Symp.*, 2001, pp. 63–68.





IEEE COMPUTER SOCIETY